Adaptive Spatial System Emergence from Community Swarm Optimization

Rawan Ghnemat, Cyrille Bertelle, Gérard H.E. Duchamp

Adaptive Spatial System Emergence from Community Swarm Optimization

Rawan Ghnemat and **Cyrille Bertelle** and **Gérard H.E. Duchamp** [1]

Abstract. This paper proposes a bottom-up approach to model spatial systems for urban dynamic and territorial management. Such approaches called micro-modelling have started with the use of cellular automata where local rules are defined and allow to implement some simulations, describing mainly dynamical diffusion processes over spatial grids. Our purpose is to deal with the detection of some emergent organizations during the simulation, using an implicit control, based on the use of genetic automata population. This automata population evolves by a swarm co-evolution, optimizing a fitness function which leads to an automatic emergence of dynamic communities.

1 SPATIAL SYSTEM EMERGENCE MODELLING

1.1 From Top-down to Bottom-up approches

Spatial systems modelling for urban dynamic or territorial management has been developped since few decades and can be classify in two main categories [2]. The first one is based on macro-modelling corresponding to a top-down approach and the second one is based on micro-modelling corresponding to a bottom-up approach.

The first approach category concerns mainly "stocks and flows" descriptions of socio-economic indicators. Ones of the main first contributors generally mentionned, are I.S.Lowry [22] and then to J.W. Forrester [12]. Lowry's model of urban system, applied to the city of Pittsburg, proposed some "integrated" model, defining flow chart between the three main indicator classes: (i) the basic sector of industrial and business activities, (ii) the householder sector and (iii) the retail sector concerning the local population. This flow chart model already deals with a mile-square decomposition similar to spatial decomposition used later as an adaptation of cellular automata grid to geographical real space. The final output of the modeling process leads to a kind of socio-economic equilibrium state. This approach finds its limit because of its static description and dynamical models are essential to understand the city evolution. Forrester proposed a dynamical modeling based on the application of industrial dynamics on urban dynamics. His model is based on non spatial stocks and flows models. Stocks are exchanged within a three income levels decomposition over housing, jobs and population. This model based on simple urban description was aimed to generate simulation and Forrester claims the benefit of computer simulation to understand the city evolution and how we can predict its evolution by the modification of guiding policies within the system.

The stocks and flows models continue to be improved and to proposed more and more details, including transportation subsystem or land market, for example. One of the most complete, called Integrated Urban Model (IUM) was proposed by Bertuglia et al. [5]. The computational complexity increase with the accuracity of the description and finally avoid to obtain reasonable estimates of the parameters. These models are more representational tools than simulation tools [2].

To buid efficient simulation models, the idea was to simplify the description, using a more global one facilitating the analytical description. From the inspiration of dynamics of population theory, some researchers proposed to build urban dynamic models from ecological modeling. The paradigm of prey-predators systems is then used to give efficient simulation tools to investigate the main feature allowing to understand the global dynamics. For example, Dendrinos and Mulally [10] use a prey-predator model, assuming that the increase of city population make decrease the economic status. The predators represent the urban population and the preys, the per capita income.

All the previous described models are based on top-down approachs to model the system dynamics. We first consider the whole phenomenon and we propose a way to split it in many sub-problems and then in stocks and flows or in different terms contituting the equational system. Another class of modeling is based on micro-modeling and bottom-up representation of the city as a collection of individual-based descriptions, behavioral rules-based description and interaction systems. From this constructive approach, we want to obtain an emergent description of the whole system or of some sub-systems included in a hierarchical process. Two complementary methodologies can be used for that and we detail them in the following paragraphs.

The first methodology consists to generate a simulation where all the components, behaviors and rules, interact over a environment, perceiving and acting on it. The environment evolving is the support of emergent properties. The cellular automata modeling deals with this kind of simulations. The basic definition of cellular automata for urban or regional modeling, for exemple, consists in the decomposition of the city, region or any geographical area in a lattice of cells. Each cell is in some state which belongs to a finite set S. At each time step, the cells change its own state according to

[1] Rawan Ghnemat, Cyrille Bertelle: LITIS - University of Le Havre, 25 rue Philippe Lebon, BP 540, 76058 Le Havre cedex, France, email: rawan.ghnemat@litislab.eu, cyrille.bertelle@litislab.eu, Gérard Duchamp: LIPN - University of Paris XIII, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse, France, ghed@lipn.univ-paris13.fr

some transition rules based on its previous state and its neighbor cells. Many works based on cellular automata, have been developped for geographical systems and urban dynamics [1, 11]. Extensions on environmental problems like water streaming are using these medels as efficient tools [20]. Cellular automata can be seen mainly as distributed tools to model diffusive penomena using rule-based systems. One of the first researchers in human sciences to propose models based on diffusive rule-based systems is T. Hägerstrand in very early period, during 50' [18] but his work itself start to be diffused over the science community more than 15 years later, when the computer development become able to implement its model in realistic studies. One of the most famous cellular-based model for social modeling is du to T. Schelling [24], describing the segregation process. But with the implementation of this model, we face to an important extension to cellular automata where we need to represent moving individuals from a simple deliberative process. The mixing of spatial data and cellular automata with autonomous entities, like agents, is here needed [8].

The second methodology to deal with emergent description in micro-modelling, consists to complete the previous approach based on simulation, by introducing some computational processes which are able to detect emergent systems or organizations. The final goal of this method is then to be able to re-introduce these emergent systems or organizations inside the simulation and manage their evolutions and their interactions with the components of the system. The re-integration of the emergent systems, during the simulation, can be explicitely expressed like in the multiscale fluid flow simulation proposed by P. Tranouez [29] or it can be implicitly expressed using a self-controled process as we will describe in the following, using genetic algorithms.

The method proposed in this paper, called community swarm optimisation, is based on a swarm intelligence process which make coevolve a population of genetic automata, using a global control of the system, implemented by a fitness function leading to the emergence of communities, as defined in the next section.

1.2 A Specific Spatial System Context: the Community

We define here, the context of the community used in the swarm intelligence process proposed in the following.

Definition 1.1 (Community operational definition)
A community is a system or an organization which is characterized by a spatial property, a behavior property and the interaction between both.

Example 1.2 *In ecology, a community is a group of plants or animals living in a specific region and interacting with one another.*

Example 1.3 *The spatial patterns generated by Schelling's segregation models [24] are some examples of communities and these spatial patterns are linked with some elementary behavioral rules implemented for each grid case. These rules describe, for each step, the movement of each individual according to its neighborhood.*

In the Community Swarm Optimization(SCO) method which we propose and describe in the following, we need to represent an efficient way to describe the behavior of each entity and we use algebraic structures called automata with multiplicities [25]. The main advantage of these automata is to be associated with algebraic operators leading to automatic computation. With these operators, we can define behavioral distances for the entities modelled with these automata. The behavioral distance is one of the major keys of this new method. Section 2 presents a review of swarm optimization methods to introduce the original one we propose in this paper. Section 3 describes the algebraic basis for the automata management used in this method. In section 4, we describe the proposed method and in section 5, we discuss some applications which can be efficiently modelled by this method, according to their own complexity.

2 SWARM OPTIMIZATION METHODS

Decentralized algorithms have been implemented for many years for various purposes. In this algorithm category, multi-agent systems can be considered as generic methods [30]. Agent-based programming deals with two main categories of agent concepts: cognitive agents and reactive agents. The first category concerns sophisticated entities able to integrate, for example, knowledge basis or communications systems. Generally, efficient computations, based on these cognitive architectures, implement few agents. The second category of agents, based on reactive architecture, is expected to be used inside numerous entity-based systems. The goals of programs using such architectures, is to deal with emergent organizations using specific algorithms called emergent computing algorithms. Swarm Intelligence is the terminology used to point out such reactive agent-based methods where each entity is built with the same basis of behavior, but reacts in autonoumous way. Swarm Optimization methods concern the problems of optimization where the computation of a function extramum is based on the concept of swarm intelligence.

Ant Colony Optimization (ACO) methods [6] is a bio-inpirated method family where the basic entities are virtual ants which cooperate to find the solution of graph-based problems, like network routing problems, for example. Using indirect communications, based on pheromon deposites over the environment (here a graph), the virtual ants react in elementary way by a probabilistic choice of path weighted with two coefficients, one comes from the problem heuristic and the other represent the pheromon rate deposit by all the ants until now. The feed-back process of the whole system over the entities is modelled by the pheromon action on the ants themselves.

Particule Swarm Optimization (PSO) is a metaheuristic method initially proposed by J. Kennedy and R. Ebenhart [19].This method is initialized with a virtual particle set which can move over the space of the solutions corresponding to a specific optimization problem. The method can be considered like an extension of a bird flocking model, like the BOIDS simulation from C.W. Reynolds [23]. In PSO algorithm, each virtual particle moves according to its current velocity, its best previous position and the best position obtained from the particles of its neighborhood. The feed-back process of the whole system over the entities is modelled by the storage of this two best positions as the result of communications between the system entities.

Other swarm optimization methods have been developped like Artificial Immune Systems [9] which is based on the metaphor of immune system as a collective intelligence process. F. Schweitzer proposes also a generic method based on distributed agents, using approaches of statistical many-particle physics [26].

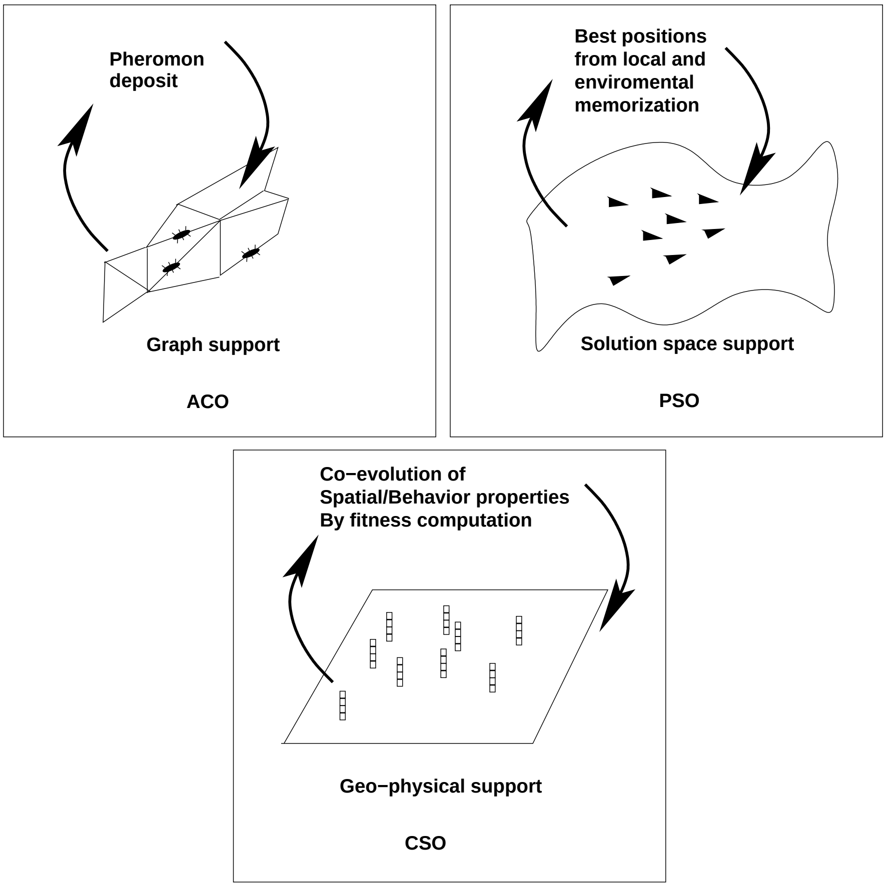


Figure 1. Support and feed-back comparison from Ant Colony Optilization (ACO), Particule Swarm Optimization (PSO) and Community Swarm Optimization (CSO)

The Communities Swarm Optimization method proposed in this paper, consists in the co-evolving of both the spatial coordinates and the behavior of each individual of a virtual population of automata. The feed-back process of the whole system over the entities is modelled by a genetic algorithm based on this co-evolving. The automata behaviors allow to define for each individual, a set of arbitrary complex transition rules. We develop the formalism needed to describe this method and the associated algorithm in the two next sections.

3 AUTOMATA-BASED AGENT MODELING AND COMPLEX SYSTEMS

3.1 Complex Systems Concepts

In this section, we give the basis of the conceptual tools which allow to extend the reactive and diffusive grid cases behavior to more sophisticated entities, using agent-based models. We propose to model the agent behavior with automata with multiplicities which are powerful algebraic stuctures.

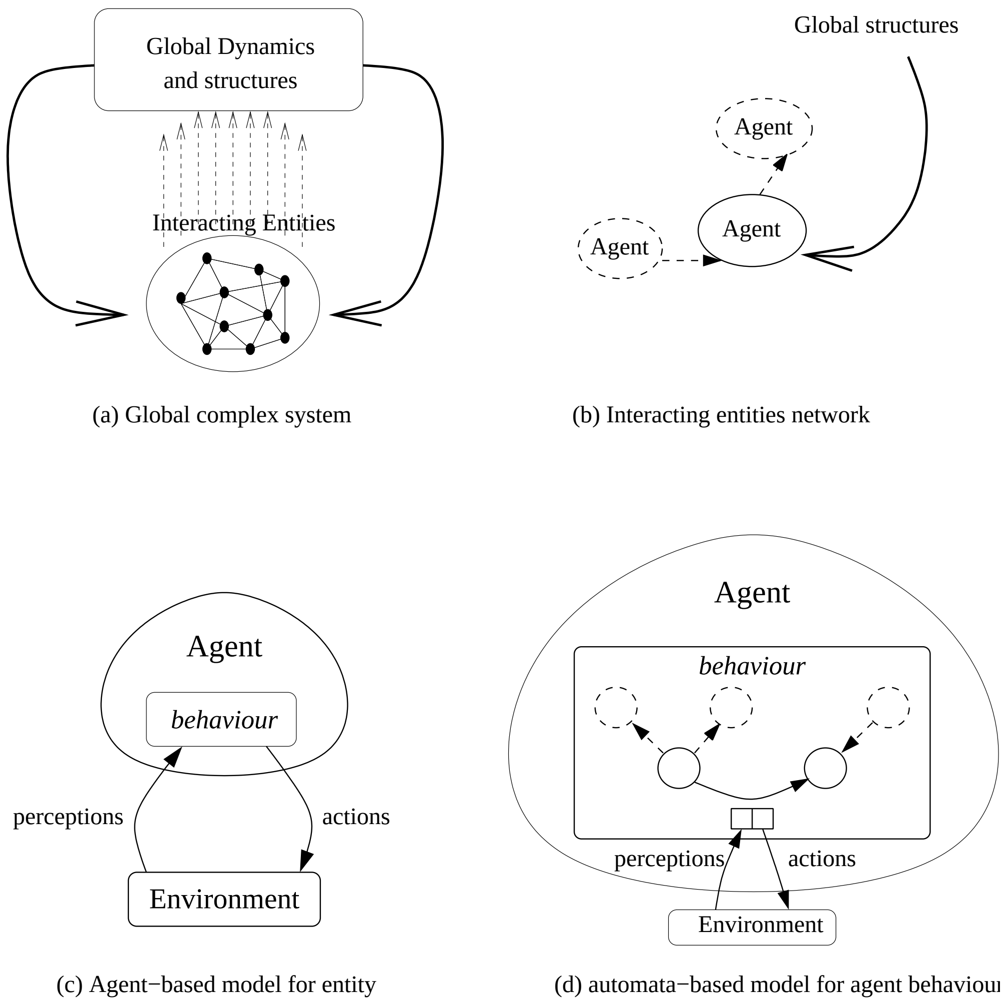
According to General System Theory [21], a complex system is composed of entities in mutual interaction and interacting with the outside environment. A system has some characteristic properties which confer its structural aspects, as schematically described in part (a) of Figure 2:

- The set elements or entities are in interactive dependance. The alteration of only one entity or one interaction reverberates on the whole system.
- A global organization emerges from interacting constitutive elements. This organization can be identified and carries its own autonomous behavior while it is in relation and dependance with its environment. The emergent organization possesses new properties that its own constitutive entities do not have.
- The global organization retro-acts over its constitutive components.

The interacting entities network as described in part (b) of Figure 2 leads each entity to perceive informations or actions from other entities or from the whole system and to act itself.

A well-adapted modeling consists of using an agent-based representation which is composed of the entity called agent as an entity which perceives and acts on an environment, using an autonomous behaviour as described in part (c) of Figure 2.

To compute a simulation composed of such entities, we need to describe the behaviour of each agent. This one can be schematically described using internal states and transition processes between these states, as described in part (d) of Figure 2. So an automaton



(a) Global complex system

(b) Interacting entities network

(c) Agent−based model for entity

(d) automata−based model for agent behaviour

Figure 2. Multi-scale complex system description: from global to individual models

with multiplicities as described in the following section is well-adapted for the agent behavior modelling.

3.2 Automata-based Modeling for Agent Behavior

An automaton with multiplicities is based on the fact that the output data of the automata with output belongs to a specific algebraic structure, a semiring, including real, complex, probabilistic, non commutative semantic outputs (transducers) [16, 27]. In that way, we will be able to build effective operations on such automata, using the power of the algebraic structures of the output data. We are also able to describe automata by means of a matrix representation with all the power of the new (i.e. with semirings) linear algebra.

Definition 3.1 (Automaton with multiplicities)
An automaton with multiplicities over an alphabet A and a semiring K is the 5-uple (A, Q, I, T, F) where

- $Q = \{S_1, S_2 \cdots S_n\}$ *is the finite set of state;*
- $I : Q \mapsto K$ *is a function over the set of states, which associates to each initial state a value of K, called entry cost, and to non-initial state a zero value ;*
- $F : Q \mapsto K$ *is a function over the set states, which associates to each final state a value of K, called final cost, and to non-final state a zero value;*
- T *is the transition function, that is* $T : Q \times A \times Q \mapsto K$ *which to a state* S_i*, a letter* a *and a state* S_j *associates a value* z *of* K *(the cost of the transition) if it exist a transition labelled with* a *from the state* S_i *to the state* S_j *and and zero otherwise.*

Remark 3.2 *We have not yet, on purpose, defined what a semiring is. Roughly it is the least structure which allows the matrix "calculus" with unit (one can think of a ring without the "minus" operation). The previous automata with multiplicities can be, equivalently, expressed by a matrix representation which is a triplet*

- $\lambda \in K^{1\times Q}$ *which is a row-vector which coefficients are* $\lambda_i = I(S_i)$*,*
- $\gamma \in K^{Q\times 1}$ *is a column-vector which coefficients are* $\gamma_i = F(S_i)$*,*
- $\mu : A^* \mapsto K^{Q\times Q}$ *is a morphism of monoids (indeed* $K^{Q\times Q}$ *is*

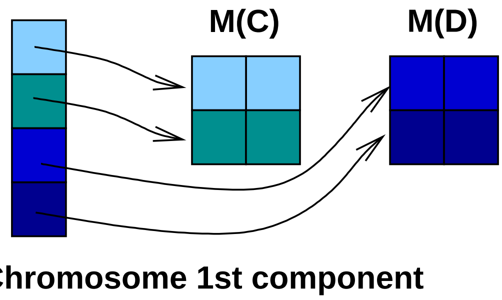


Figure 3. Chromosome first component building from the matrix rows of the linear representation of an automaton over the alphabet {C,D}

endowed with the product of matrices) such that the coefficient on the q_ith row and q_jth column of $\mu(a)$ is $T(q_i, a, q_j)$

Definition 3.3 (Automata-Based Agent Behavior)
We represent the agent behavior by automata with multiplicities (A, Q, I, T, F) over a semiring K:

- *The agent behavior is composed of a states set Q and of rule-based transitions between them. These transitions are represented by T; I and F represent the initial and final transitions;*
- *Alphabet A corresponds to the agent perceptions set;*
- *The semiring K is the set of agent actions, eventually associated to a probabilistic value which is the action realization probability (as defined in [13]).*

3.3 Agent Behavior Metric Space

The main advantage of automata-based agent modelling is their efficient operators. We deal is this paragraph with an innovative way to define behavioral semi-distance as the essential key of self-organization processus proposed later.

Definition 3.4 (Evaluation function for automata-based behavior)
Let x an agent whom behavior is defined by A, an automaton with multiplicities over the semiring K, we define the evaluation function $e(x)$ by:

$$e(x) = V(A)$$

where $V(A)$ stands for the vector of all coefficients of (λ, μ, γ), the linear representation of A, defined in remark 3.2.

Definition 3.5 (Behavioral semi-distance)
Let x and y two agents and $e(x)$ and $e(y)$ their respective evaluations as described in the previous definition 3.4. We define $d(x, y)$ a semi-distance or pseudometrics [2] between the two agents x and y as

$$d(x, y) = ||e(x) - e(y)||$$

a vector norm of the difference of their evaluations.

Remark 3.6 *In this paper, we propose a simple computation for the behavioral semi-distance. It is possible to define other behavioral semi-distances who can allow to introduce specific similarities on specific path inside the automata and not the complete description of the automata as used here. This process consists in defining specific initial and final states and compute all the successfull paths between them [25] . We will not develop this extension in this paper.*

[2] see [7] ch IX

4 COMMUNITY SWARM OPTIMIZATION ALGORITHM

4.1 Spatial Automata-based Agent

Definition 4.1 (Spatial Automata-Based Agent)
A spatial automata-based agent is defined by its structural representation:

- *An automaton with multiplicities corresponding to its behavior as a whole processus managing its perceptions and its actions over its environment. They include its communication capabilities and so its social behavior;*
- *A spatial location defined on some specific metric space.*

4.2 Genetic Operators on Automata Population

We consider in the following, a population of spatial automata-based agents, each of them is represented by a chromosome, following the genetic algorithm basis. We define the chromosome for each spatial automata-based agent as a couple of two sequences:

- the sequence of all the rows of the matrices of the linear representation of the automata. The matrices, associated to each letter from the alphabet of the agent perceptions, are linearly ordered by this alphabet and we order all the rows following these matrices order [4]. The figure 3 describes how this sequence is created from a linear representation of two matrices;
- the sequence of all its spatial coordinates.

In the following, genetic algorithms are going to generate new automata containing possibly new transitions from the ones included in the initial automata.

The genetic algorithm over the population of automata with multiplicities follows a reproduction iteration broken up in two steps [17]:

- *Reproduction (Duplication and Crossing-over)*: This operator is a combination of the standard duplication and crossing-over genetic operators. For each couple of spatial automata (called the parents), we generate two new spatial automata (called the children) as the result of the chromosome crossings and we keep, without change, the parent spatial automata. To operate for the crossing-over operation, we have to compute the automata of the behaviors of the two children. For this purpose, we consider a sequence of rows for each matrix of the linear representation of one of the two parents and we make a permutation on these chosen sequences of rows between the analogue matrix rows of the other parent.

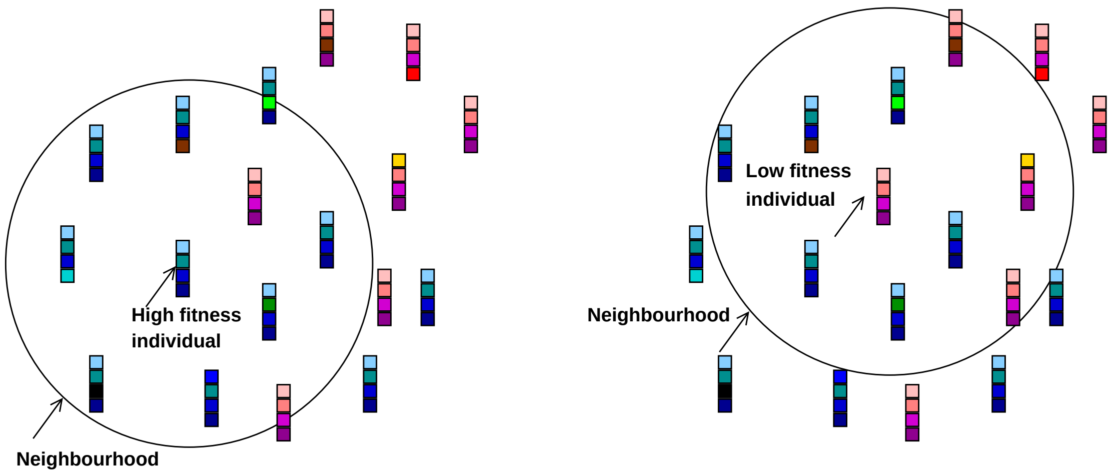


Figure 4. Fitness evaluation for community detection

- *Mutation*: This operator deals only with the linear representation of the spatial automata-based agent. With a low probability, each matrix row from this linear representation is randomly chosen and a sequence of new values is given for this row (respecting some constraints if exist, like probabilistic values [4]).

Remark 4.2 *The fitness is not defined at this level of abstract formulation, but it is defined corresponding to the context for which the automaton is a model, as we will do in the next section.*

4.3 Adaptive Processus to Implement Community Detection

The community detection is based on a genetic algorithm over a population of spatial automata-based agents. The formation of the community is the result of the population evolution crossing by a selection process computed with the fitness function defined in the following.

For this computation, we deal with two distances defined on agent set. The first is the spatial distance associated to the agent spatial location and the second is the behavioral semi-distance defined in the definition 3.5.

Definition 4.3 Community clustering and detection fitness *Let V_x a neighbourhood of the agent x, relatively to its spatial location. We define $f(x)$ the agent fitness of the agent x as :*

$$f(x) = \begin{cases} \dfrac{card(V_x)}{\displaystyle\sum_{y_i \in V_x} d(x,y_i)^2} & \textit{if } \displaystyle\sum_{y_i \in V_x} d(x,y_i)^2 \neq 0 \\ \infty & \textit{otherwise} \end{cases}$$

where $d(x,y)$ is the behavioral semi-distance between the two agents x and y.

On the figure 4, we represent an automata population where each automata is a colored chain representing its chromosom. Automata with similar colored chain must ne understanding as similar behavioral automata. In the left part of the figure, we focuss on one high fitness individual after computing its spatial neighbourhood and observing the behavioral similarity of all the automata included in this neighbourhood. In the right part of the figure, composed of the same population, we focuss on a low fitness individual, having not similar behavior with the other automata belonging to its neighbourhood.

The genetic evolution of the spatial automata-based agents leads to a self-organization which creates a clustering of the agents set in such way that each cluster contains agents of similar behavior. During the evaluation process, genetic algorithms can be turned such that individuals *outside communities* be attracted to them. The center of the clusters, the size of the clusters and the behavior of the agents in the center of each cluster are the result of the overall genetic processus which generates self-organization communities.

4.4 General Community Swarm Optimization Algorithm

Community Swarm Optimization (CSO) algorithm needs a initial step description which is the major issue of the modelling process. The way of going from the problem formulation to the initial spatial automata-based agents must be realized with accuracity. The formal description of the methodology to use, for this initial step, is described in Algorithm 1.

The core of the CSO algorithm is described by the iterative scheme defined in the Algorithm 2.

5 Conclusion and Perspective

In this paper, we describe Community Swarm Optimization (CSO) method which can be described as a swarm intelligence process. With the comparison of other methods from its category (Ant

Algorithm 1: Methodology to model the initial step of CSO

1. Problem formulation by the definition of a set of transition rules ;
2. Building of the behavioral automata based on the previous set of transition rules, describing the sequences and the context of their applications ;
3. Discretization of the spatial domain, according to its topological properties (Cellular automaton, network or graph, Geographical Information System) with the spatial location of the initial virtual population of spatial automata-based agents;

Algorithm 2: Iteratice scheme of CSO

Building the initial virtual population of the spatial automata-based agents (following the methodology of Algorithm (1)) ;
repeat
    for *Each couple of individuals in the population* **do**
        Reproduction step generating 2 new children as described in the section (4.2) ;
        Mutation step as described in the section (4.2) ;
    Selection of the half population of the individuals corresponding to the highest values of the agent fitness described in section (4.3) ;
until *(the sum of the fitness values of the whole population reachs a threshold)* or *(the maximum iteration number is reached)* ;

Colony Optimization and Particule Swarm Optimization), CSO differs mainly on the modelling purpose. CSO deals with transition rules included in data structures (automata with multiplicities) for which algebraic operators allow to implement automatic computation for self-organizational phenomena.

This method is expected to be used for adaptive spatial system emergence modelling. The swarm intelligence method proposed here manage artificial population leading to the emergent formation of communities. The genetic automata which compose this artificial population, allow the emergent communities to be self-controled using the fitness of the genetic process. This fitness express a global control over the community, using similarity evaluation on spatial neighborhood. This process mix individual representation and community system emergence: it is the formalization of a multi-scale description where the micro-macro interactions are implicit express by the genetic control of the emergent communities. The application expected to be model by this process, concern for exemple urban dynamic where quarters or city centers can emerge or evolve from the citizen behavior like in the gentrification problem.

REFERENCES

[1] M. Batty, Y. Xie, From cells to cities, Environment and Planning B, 21 (1994) 531-548.

[2] I. Benenson, Modeling Population Dynamics in the City: from a regional to a Multi-Agent Approach, Discrete Dynamics in Nature and Society 3 (1999) 149-170.

[3] Benenson, I., Torrens, P.M. (2004) *Geosimulation - Automata-based modeling of urban phenomena*, Wiley.

[4] Bertelle, C., Flouret, M., Jay, V., Olivier, D., Ponty, J.-L. (2001) *Genetic algorithms on automata with multiplicities for adaptive agent behaviour in emergent organizations* In *SCI'2001*, Orlando, Florida, USA, 22-25th July 2001.

[5] C.S. Bertuglia, S. Occelli, G.A. Rabino and R. Tadei, An integrated urban model, in: Bertuglia, Leornardi, Occelli, Rabino, Tadei and Wilson (eds), Urban Systems: contempory approach to modelling, Croom Helm, London, pp 178-191.

[6] Bonabeau, E., Dorigo, M., Theraulaz, G. (1999) *Swam Intelligence, from natural to artificial systems*, a volume in the Santa Fe Institute Studies in the Sciences of Complexity, Oxford University Press.

[7] Bourbaki, N. (1998) *Elements of Mathematics: General Topology*, Chapters 5-10, Springer-Verlag Telos.

[8] E. Daude, P. Langlois, Introduction à la modélisation multi-agents des systèmes complexes en géographie, 357-441, in F. Amblard and D. Phan (eds) Modélisation et simulation multi-agents, Lavoisier, Herms Science, Paris, 2006.

[9] De Castro, L.N., Timmis, J. (2002) *Artificial immune system: a new computational approach*, Springer-Verlag, London, U.K.

[10] D. Dendrinos, H. Mullaly, Urban Evolution: Studies in the Mathematical Ecology of the Cities. Oxford, Oxford University Press, 1985.

[11] E. Dubos-Paillard, Y. Guermond, P. Langlois, Analyse de l'évolution urbaine par automate cellulaire. Le modèle SpaCelle, L'espace géographique 32 (2003/4), 357-378.

[12] J.W. Forrester, Urban Dynamics, Cambridge, MA, MIT Press, 1969.

[13] Ghnemat, R., Oqeili, S., Bertelle, C., Duchamp, G.H.E. (2006) *Automata-Based Adaptive Behavior for Economic Modelling Using Game Theory*, in Aziz-Alaoui, M.A., Bertelle, C., *Emergent Properties in Natural and Artificial Dynamical Systems*, Springer.

[14] Ghnemat, R., Bertelle, C., Duchamp, G.H.E. (2007) *Adaptive Automata Community Detection and Clustering, a generic methodology*, in *Proceedings of World Congress on Engineering 2007, International Conference of Computational Intelligence and Intelligent Systems*, pp 25-30, London, U.K., 2-4th July 2007.

[15] Ghnemat, R., Bertelle, C., Duchamp, G.H.E. (2007) *Self-organized communities modelling by genetic spatial automata*, EPNACS'07 satellite conference within ECCS'2007, Dresden, Germany.

[16] Golan, J.S. (1999) *Power algebras over semirings*, Kluwer Academic Publishers.

[17] Goldberg, D.E. (1989) *Genetic Algorithms*, Addison-Wesley.

[18] T. Hagerstand, Innovation Diffusion as a spatial process (translated in English by Allan Pred, 1967) Lund, Sweeden, 1953. Chicago, University of Chicago Press.

[19] Kennedy, J., Eberhart, R.C. (1995) *Particle Swarm Optimization* In *Proceedings of IEEE International Conference on Neural Networks (Perth, Australia)*, IEEE Service Center, Piscataway, NJ, 5(3), pp 1942-1948.

[20] P. Langlois, D. Delahaye, RuiCells, automate cellulaire pour la simulation du ruissellement de surface, Revue internationale de Gomatique, Vol. 12/4 (2002) 461-487.

[21] Le Moigne, J.-L. (1999) *La modélisation des systèmes complexes*, Dunod.

[22] I.S. Lowry, A model of metropolis, Santa Monica, CA, the RAND Corporation: 136, 1964.

[23] Reynolds, C.W. (1987) *Flocks, Herds and Schools: a distributed behavioral model* In *Computer Graphics*, 21(4) (SIGGRAPH'87 Conference Proceedings), pp 25-34.

[24] Schelling, T.C. (1971) *Dynamic Models of Segregation*, Journal of Mathematical Sociology, Vol. 1:143-186.

[25] Schützenberger, M.P. (1961) *On the definition of a family of automata*, Information and Control, Vol. 4:245-270.

[26] Schweitzer, F. (2003) *Brownian Agents and Active Particles*, Springer.

[27] Stanley, R.P. (1999) *Enumerative combinatorics*, Cambridge University Press.

[28] Strout, N. (2006) *What is Agent Analyst*, The 2006 ESRI International User Conference, San Diego.

[29] P. Tranouez, C. Bertelle, D. Olivier, Changing levels of description in a fluid flow simulation, in M.A. Aziz-Alaoui and C. Bertelle (eds) Emergent Properties in Natural and Artificial Dynamics Systems, Springer complexity "Understanding Complex Systems" series, Springer, Berlin Heidelberg, 2006.

[30] Weiss, G. (ed.) (1999) *Multiagent Systems*, MIT Press.